

## מבני נתונים – תרגיל 3

תאריך פרסום: 19.4.16

תאריך הגשה: 15.5.16

מרצה ומתרגלים אחראים: פרופ' שלומי דולב, תומר כהן, עומרית פילצר.

### נושאי העבודה:

- תכנון מבנה נתונים יעיל לאלגוריתם.
- ניתוח זמן הריצה של החלקים השונים באלגוריתם.
- מימוש המבנה ב Java ובדיקתו.

טיפ: קראו תחילה את כל התרגיל, מתחילתו ועד סופו, והתחילו לעבוד רק לאחר שהבנתם את כל הפרטים.

### רקע:

דיפי והלמן יצרו בפעם הראשונה שיטה ליצירת סוד משותף בין שני משתתפים בנוכחות מאזין. האלגוריתם שלהם משתמש בפונקציה חד-כיוונית (שקשה להפוך אותה, כדוגמא אינטואיטיבית אפשר לחשוב על העלאה בריבוע לעומת הוצאת שורש ריבועי, בעוד שקל להעלות בריבוע יותר קשה (ידינית) להוציא שורש). גישה חדשה זו שינתה את פני ההצפנה למה שנקרא היום קריפטוגרפיה מודרנית, ולמעשה משרתת אותנו ביום-יום לתקשורת מאובטחת באינטרנט. הבעיה היא שאין הוכחה שהפונקציה שמשמשת אותנו היום הינה באמת חד-כיוונית. יותר מכך, אין הוכחה שקיימת פונקציה חד-כיוונית בכלל. השנה דיפי והלמן זכו בפרס טיורינג (המקביל לפרס נובל במדעי המחשב) על ההמצאה שלהם. אחד מהם הזכיר את ראלף מרקל כמי שהיה שותף להמצאה של שיטה ליצירת סוד על סמך קושי חישובי.

תרגיל הבית יתרכז בשיטה שהציג מרקל (שעבורה ניתן להוכיח קושי, כפי שתדגימו) הרעיון באחת הגרסאות שלו מתבסס על יצירת חידות בצורה רנדומית על ידי אחד המשתתפים, בחירת רנדומית של חידה מתוך אלה שהגיעו ופתרונה על ידי המשתתף השני, ושליחת חלק מהפתרון לראשון, בעוד שהחלק השני משמש כסוד המשותף שנוצר. מי שמקשיב לחילופי האינפורמציה צריך לפתור את כל החידות (או כמחציתן בממוצע) עד שיגיע לחצי הפתרון שנשלח כתשובה כדי לדעת מהו הסוד שהוחלט לשמש את שני המשתתפים להצפנת התקשורת ביניהם, עד אז הם יתקשרו בביטחה (מסוימת).

לדוגמא, נניח שאליס רוצה להעביר לבוב מידע סודי כלשהו (מפתח הצפנה למשל), אך קו התקשורת שלה איתו הוא ציבורי, כלומר ניתן לצותת לו. היא מכינה חידות רבות, כאשר כל חידה היא הצפנה של מספר סידורי ומפתח הצפנה סודי, ושולחת אותן לבוב. בוב בוחר חידה אחת, פותר אותה, מחלץ את המספר הסידורי ומפתח ההצפנה, ושולח לאליס את המספר הסידורי בלבד. אליס, שהכינה את החידות בעצמה, יודעת באמצעות המספר הסידורי איזו מהחידות פוענחה, ולכן יודעת באיזה מפתח הצפנה בחר בוב. כעת הם יכולים לתקשר ביניהם באופן בטוח ע"י שימוש באותו מפתח הצפנה.

המצותתת איב מסוגלת לראות את כל החידות ויכולה לפענח בדיוק כמו בוב כל חידה שתבחר, אך היא אינה יודעת איזו מהחידות בוב פענח והמספר הסידורי מוסתר בתוך החידה. לכן עליה לנסות לפתור את כל החידות (או לפחות מחציתן במקרה הממוצע) כדי למצוא את המספר הסידורי שהחזיר בוב לאליס, ואז לחלץ את המפתח שהם שיתפו ביניהם.

## רעיון המימוש:

בתרגיל נניח שכל אחת מ- $K$  החידות שאלים יוצרת היא שני מערכים בגודל  $N$  של מספרים מהתחום  $[0, N^4]$ .

אליס בוחרת  $K$  זוגות מערכים אקראיים כאלה וממיינת אותם. למעשה, היא יכולה לבנות רנדומית מערכים ממוינים ורק אחר כך ל"בלגן" אותם רנדומית, וכך לחסוך את הצורך למיין אותם (ראו פירוט בהמשך).

לאחר מכן, בכל מערך היא מבצעת פעולת XOR על הביטים של כל אחד מהמספרים במערך, ומקבלת סדרה של  $N$  ביטים המתאימה למערך הממוין. כלומר, לכל חידה מתאימות שתי סדרות בנות  $N$  ביטים כל אחת. הסדרה הראשונה משמשת כתשובה אפשרית של המקבל (מספר סידורי של החידה), והסדרה השנייה משמשת כמפתח הצפנה הסודי.

**שימו!** יתכן שתהיה אותה סידרה של ביטים בשני מערכים שונים, אך ההתסברות לכך קטנה אקספוננציאלית עם אורך המערכים. במקרה שמתקבל מערך עם סדרת ביטים שהופיעה כבר, אליס מתעלמת ממנו ומנסה שוב. התעלמו מתוספת זו לזמן הריצה בתשובתכם בחלק ב' של התרגיל.

בוב בוחר אחת מ- $K$  החידות, ממין כל אחד מזוג המערכים המתאימים לה, מבצע פעולת XOR על הביטים של כל אחד מהמספרים במערכים הממוינים ומקבל שתי סדרות של  $N$  ביטים. לאחר מכן הוא שולח לאליס את הסדרה הראשונה (המספר הסידורי) ומשתמש בסדרה השנייה כמפתח ההצפנה הסודי. כמובן שהבטיחות הינה זמנית כיוון שאיב תצליח למיין לבסוף גם את המערך שבוב בחר רנדומית, אך אולי זה יהיה מאוחר מידי.

למעשה, ניתן להשתמש במפתח הסודי כדי להצפין (בשימוש במפתח הראשון כ-seed ליצירה של סידרה פסאודו-אקראית בשני הצדדים) סידרה חדשה של מערכים, שתשמש לבחירת מפתח משותף בטוח יותר, וכך הלאה, ורק אז לשלוח מידע משמעותי מוצפן באמצעות המפתח בין שני הצדדים.

בתרגיל תממשו את התוכניות של אליס ובוב כך שהם יצרו מפתח משותף על פי בקשה. בנוסף, תממשו את התוכנית של איב כהליך שמנסה למצוא את המפתח בצורה המהירה ביותר.

## דוגמא:

עבור  $N = 2, K = 3$ .

- אליס יוצרת את החידות הבאות:
  1.  $[13,7], [5,9]$
  2.  $[11,15], [10,5]$
  3.  $[9,3], [12,2]$
- בוב בוחר את חידה מספר 2, ופותר אותה כך:
  - הוא ממין את המערכים של חידה מספר 2 בלבד, ומקבל:  $[11,15], [5,10]$
  - ביצוג בינארי, המספרים הם:  $[1011,1111], [101,1010]$
  - הסדרות המתאימות הן: 10, 00
  - בוב שולח לאליס את הסדרה 10, והמפתח הסודי שהוא שומר הוא: 00
- אליס כבר יודעת את הפתרון לכל החידות, וכאשר היא מקבלת את תשובתו של בוב היא מיד יודעת למצוא את המפתח הסודי המתאים.
- לאיב אין את הפתרונות של החידות, והיא צריכה לפתור אותן אחת-אחת עד שתגלה מיהי החידה שהמספר הסידורי שלה הוא 10.

## חלק א' - מימוש השיטה:

ממשו את הפונקציות הבאות בקבצים המצורפים לתרגיל. ניתן להוסיף פונקציות ומחלקות משלכם.

1. המחלקה Alice:

- a. מכילה שדה בשם Puzzles.
- b. הפונקציה createPuzzles מקבלת  $N$  ו- $K$ , מייצרת את אוסף החידות (מערכים) בצורה רנדומית, ושומרת אותן בשדה Puzzles.
- c. הפונקציה getPuzzles (נתונה לכם) מחזירה העתקה עמוקה של השדה Puzzles.
- d. הפונקציה findKey מקבלת מספר סידורי (מחרוזת באורך  $N$ ), ומחזירה את המפתח הסודי (מחרוזת באורך  $N$ ), המתאים לחידה המכילה את אותו מספר הסידורי. בנוסף הפונקציה תחזיר את מספר ההשוואות בין מספרים סידוריים של חידות שביצעה אליס ע"מ למצוא את המפתח הסודי המתאים למספר הסידורי שקיבלה.

2. המחלקה Bob:

- a. הפונקציה solvePuzzle מקבלת חידה (זוג מערכים), פותרת אותה, ומחזירה את המספר הסידורי של החידה ואת המפתח הסודי המתאים לה (מחרוזת באורך  $N$ ).
- b. הפונקציה choosePuzzle מקבלת מערך של חידות, בוחרת באקראי חידה מתוכן, קוראת לפונקציה solvePuzzle ומחזירה את המספר הסידורי של החידה ואת המפתח הסודי המתאים לה (מחרוזת באורך  $N$ ).

3. המחלקה Eve:

- a. הפונקציה solvePuzzle מקבלת חידה (זוג מערכים), פותרת אותה, ומחזירה את המספר הסידורי של החידה ואת המפתח הסודי המתאים לה (מחרוזת באורך  $N$ ).
- b. הפונקציה findKey מקבלת מערך של חידות ומספר סידורי (מחרוזת באורך  $N$ ), ומחזירה את המפתח הסודי (מחרוזת באורך  $N$ ), המתאים לחידה המכילה את אותו מספר הסידורי. בנוסף הפונקציה תחזיר את מספר החידות שהיא בדקה עד שמצאה את זו שבו בחר.

## יצירת המערכים האקראיים:

אליס רוצה לבנות רנדומית מערכים ממוינים, ורק אחר כך ל"בלגן" אותם רנדומית, וכך לחסוך את הצורך למיין אותם. היא עושה זאת באופן הבא:

תחילה היא מגרילה כל איבר  $i$  במערך מהתחום  $[(i-1)N^3, iN^3]$ , כלומר כל המספרים מ- $(i-1)N^3$  (כולל), ועד  $iN^3$  (לא כולל).

לאחר מכן היא "מבלגנת" את המערך בעזרת הפונקציה הבאה:

```
RandomeShuffle(Arr) {  
    for ( $i = 0$  to  $N - 1$ )  
        swap(Arr[i], A(Random( $i$ ,  $N - 1$ )))  
}
```

## מיון המערכים האקראיים:

ע"מ לפתור כל חידה, בוב ואיב צריכים למיין תחילה את המערך הנתון. למדתם שניתן למיין מערך בגודל  $N$  בזמן  $O(N \log N)$ . בהנחה שהם יודעים באיזו שיטה משתמשת אליס ע"מ לבנות את המערכים האקראיים שלה, האם יוכלו לעשות זאת בזמן  $O(N)$ ?

במימוש שלכם לבוב ואיב נסו למצוא דרך לפתור את החידות בצורה היעילה ביותר האפשרית.

**שימו!** ♥ מיון בזמן  $O(N \log N)$  מספיק כדי לזכות אתכם בכל הנקודות עבור הסעיף הזה. מיון בזמן  $O(N)$  יזכה אתכם ב **בונוס**. אם בחרתם במיון בזמן  $O(N)$ , פרטו כיצד עשיתם זאת בחלק ב' של התרגיל.

### שמירת אוסף הפתרונות:

אליס מעוניינת לשמור את אוסף הפתרונות של החידות שיצרה לפי מספר סידורי, ע"מ שכאשר תקבל מבוב את המספר הסידורי שבחר תוכל למצוא בקלות וביעילות את המפתח הסודי המתאים לו. שימו לב שישנן מספר אפשרויות לעשות זאת באופן יעיל. לא מזמן אליס למדה בקורס מבני נתונים על עצי AVL, והיא רוצה לנסות את כוחה במימוש עץ כזה, על מנת ללמוד ולהפנים טוב יותר כיצד הוא פועל.

**שימו!** ♥ בסעיף זה, מציאת המספר הסידורי בזמן  $O(\log K)$  מספיק כדי לזכות אתכם בכל הנקודות עבור הסעיף. שימוש בעץ AVL שתממשו בעצמכם בהתאמה לתרגיל יזכה אתכם ב **בונוס**.

### הרצת התוכנית:

המחלקה MerklePuzzles מיועדת לבדיקת המימוש שלכם. היא מכילה את הפקודות הבאות שתוכלו להריץ:

- הפקודה `Alice_createPuzzles[K][N]` תריץ את הפונקציה `createPuzzles` של אליס עם הפרמטרים  $K$  ו- $N$ , ותציג את הזמן שלקח לה לעשות זאת.
- הפקודה `Bob_choosePuzzle` תריץ את הפונקציה `choosePuzzle` של בוב עם המערך המתקבל מהפונקציה `getPuzzles` של אליס, ותציג את המספר הסידורי והמפתח הסודי המתאימים לחידה שבחר בוב, ואת הזמן שלקח לו לעשות זאת.
- הפקודה `Alice_findKey[number]` תריץ את הפונקציה `findKey` של אליס עם המספר הסידורי `number` כקלט, ותציג את המפתח המתאים לחידה, את מספר פעולות ההשוואה שביצעה, ואת הזמן שלקח לה לעשות זאת.
- הפקודה `Eve_findKey[number]` תריץ את הפונקציה `findKey` של איב עם המערך המתקבל מהפונקציה `getPuzzles` של אליס, והמספר הסידורי `number` כקלט, ותציג את המפתח המתאים לחידה ואת הזמן שלקח לה לעשות זאת.
- הפקודה `fullTest[K][N]` תריץ את הפקודות הבאות לפי הסדר:
  1. הפקודה `Alice_createPuzzles[K][N]`.
  2. הפקודה `Bob_choosePuzzle`.
  3. הפקודות `Alice_findKey` ו-`Eve_findKey`, עם המספר הסידורי שבחר בוב.

הריצו את התוכנית ובחרו  $K$  ו- $N$  כרצונכם עבור פקודת `Alice_createPuzzles`, לאחר מכן הריצו את שאר הפקודות כרצונכם. שימו לב שיש להריץ את פקודת `Alice_createPuzzles` קודם, ע"מ לייצר את אוסף החידות.

## חלק ב' – סיכום התרגיל:

ענו על השאלות הבאות וצרפו את קובץ התשובות ל־zip. המסמך צריך להיות מוקלד ובפורמט PDF.

- א. תארו בקצרה את המימוש שלכם: באילו מבני נתונים השתמשתם? למה מיועד כל מבנה? התיאור צריך להיות במילים, אין צורך לכתוב את הקוד מחדש.
- ב. הסבירו בקצרה את זמן הריצה של כל אחת מן הפעולות שנדרשתם לממש – הסבירו כיצד אתם עומדים בזמן הריצה לפי המימוש שלכם. גם במקרה זה עליכם להסביר במילים ולא בחלקים מהקוד.
  1. מה יהיה זמן הריצה של אליס ליצירת כל החידות?
  2. מה יהיה זמן הריצה של בוב עבור פתרון חידה אחת?
  3. מה יהיה זמן הריצה של אליס לגילוי החידה שבחר בוב?
  4. מה יהיה זמן הריצה של איב לגילוי החידה שבחר בוב? פרטו מה הוא יהיה בממוצע, ומה במקרה הגרוע.
- ג. הריצו את התוכנית עם הפרמטרים שבטבלה הבאה, 10 פעמים לכל זוג פרמטרים. השלימו את הטבלה לפי התוצאות שקיבלתם, והשלימו את שלושת השורות האחרונות עם  $N$  ו- $K$  כרצונכם:

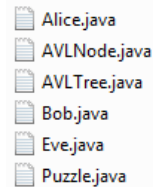
$K$	$N$	מספר פעולות ההשוואה שביצעה אליס בממוצע	הזמן שלקח לאליס בממוצע	מספר החידות שבדקה איב בממוצע	הזמן שלקח לאיב בממוצע
5000	100				
10000	100				
20000	100				

באופן תיאורטי (התעלמו מאילוצי חומרה), האם אליס תמיד תצליח לגלות באיזו חידה בחר בוב לפני שאיב תגלה זאת? אם כן, מדוע? אם לא, מתי זה יכול לקרות? הסבירו.

- ד. בהינתן  $N$ , מהו ה- $K$  המקסימלי שיוכלו אליס ובוב לבחור? האם במקרה זה לאיב יהיה הכי קשה לגלות את הסוד? מה יהיה היחס בין הזמן (במקרה הגרוע) שיקח לאליס לגלות מהי החידה לבין הזמן (במקרה הגרוע) שיקח לאיב לגלות מהי החידה (כלומר היחס בין הפונקציות findKey של אליס ואיב)?

## הערות חשובות ודרישות הגשה:

1. ניתן להגיש את העבודה ביחידים או בזוגות לפי רצונכם, אין צורך לבקש אישור מיוחד להגשה ביחידים.
2. ניתן להניח שהקלט יהיה תקין. לא תקבלו מתכנית הבדיקה שלנו ערך null או טקסט ריק כפרמטר לאף אחת מהפונקציות בממשק.
3. את העבודה יש להגיש באמצעות ה-Submission System. עליכם להגיש קובץ zip בשם assignment3.zip המכיל בתוכו:
  1. תיקיית src ובה קבצי הג'אווה של העבודה:



Alice.java  
AVLNode.java  
AVLTree.java  
Bob.java  
Eve.java  
Puzzle.java

2. מסמך PDF לפי הנדרש, בשם "PartB.pdf".
4. סביבת העבודה בה תיבדקנה העבודות הינה JavaSE-1.7/8 9 עליכם לדאוג כי עבודותיכם יתקמפלו וירוצו בסביבת eclipse תחת גרסאות Java הנזכרות לעיל.
5. עבודות שלא יתקמפלו – יקבלו ציון 0.
6. עבודותיכם יבדקו באמצעות כלי בדיקה אוטומטים הבודקים קורלציה בין עבודות. נא לא להעתיק! להזכירכם, המחלקה רואה בחומרה רבה העתקות.
7. נרצה לראות קוד מתועד, מתוכנן היטב ויעיל שמייצג הבנה.